

**AMENDMENTS TO THE SPECIFICATION**

1. Please amend paragraph [0047] as follows:

[0047] In the above-described structure of the standardized command control table 34, a standardized command “typedef enum” such as “DIA\_ COMMAND\_ID\_SDH\_E” exists as “DIA\_ COMMAND\_ID\_PDH\_E”, “DIA\_ COMMAND\_ID\_ATM\_E”, *etc.* on a field basis. Various command functions can be arbitrarily added to the standardized command control table 34. The user of the higher-order application can arbitrarily call a corresponding device driver through the standardized command control table 34. Although a device driver is changed, a corresponding standardized command remains as it is. Only a function pointer is varied when the corresponding device driver carries out a corresponding command. The function pointer is provided to the DIA hierarchy 22 by the device driver when a device is initialized. The command is called through an API “Dia\_Control” by the user of the higher-order application. When the user of the higher-order application calls a specific control command ~~the API “Dia\_Control”~~ through the API “Dia\_Control” ~~a specific control command~~, only a device driver positioned below the DIA hierarchy 22 is varied.

2. Please amend paragraph [0050] as follows:

[0050] An event table 38 shown in Fig. 5 is a structure located at a position pointed by the pointer of “\*pEventTable” of a corresponding DCB being generated only if necessary. When the event table is not used, it is indicated as a null. The event table 38 includes an event ID of “EventId” and an event list structure pointer of “\*pEventList”. Each event list is a linked list as shown in Fig. 9. Accordingly, a plurality of places can be notified of one event and a call-back function of “Call-back Fn” can be called. The event table 38 initially includes the event ID of “EventId” and the pointer of “\*pEventList” is initially pointed as a null. When the user of the higher-order application 20 uses a certain corresponding event, the user of the higher-order application 20 is able to register and use the event by connecting to “\*pEventList”, ~~can be connected to~~ a pointer of an event list structure using an API “Dia\_Register”.